**WHAT IS CLAIMED IS:**

1. A method of testing error correction/detection logic, the method comprising:

creating an initial data bit combination having n bits, wherein each data bit in the initial data bit combination has a same logical value as each other data bit in the initial data bit combination;

shifting a first bit having an different logical value than the same logical value across the initial data bit combination, wherein each time the first bit is shifted, one of n data bit combinations is generated;

providing each of the n data bit combinations to the error detection/correction logic;

in response to said providing, the error detection/correction logic generating a set of check bits for each of the n data bit combinations;

comparing the set of check bits generated by the error correction/detection logic with a known correct set of check bits for each of the n data bit combinations; and

dependent on an outcome of said comparing, generating an indication of whether the error detection/correction logic correctly generated the set of check bits.

2. The method of claim 1, wherein the error detection/correction logic comprises a Hamming code encoder and a Hamming code decoder.

3. The method of claim 1, further comprising:

providing a set of m+1 test code words to the error correction/detection logic, wherein each code word has m bits, wherein a first test code word in the set of m+1 test code words is a correct code word, wherein each test code word other than the first test code word comprises a single-bit error at a different bit position within the code word than each other test code word;

in response to said providing, the error correction/detection logic decoding the set of m+1 test code words; and

verifying that the error correction/detection logic correctly decoded each of the m+1 test code words.

4. The method of claim 3, wherein each bit in the first test code word has a same logical value, and wherein each test code word other than the first test code word comprises a bit having an opposite logical value at the different bit position.

5. The method of claim 4, wherein said providing a set of m+1 test code words comprises shifting the bit having the opposite logical value across the first test code word, wherein each time the bit is shifted, one of the test code words other than the first test code word is created.

6. The method of claim 3, wherein said verifying that the error correction/detection logic correctly decoded each of the m+1 test code words comprises verifying that the error correction/detection logic detects each of the single-bit errors.

7. The method of claim 3, wherein said verifying that the error correction/detection logic correctly decoded each of the m+1 test code words comprises verifying that the error correction/detection logic corrects each of the single-bit errors that occurs in a data bit within each of the m+1 test code words.

8. The method of claim 3, wherein said verifying that the error correction/detection

logic correctly decoded each of the m+1 test code words comprises verifying that the error correction/detection logic indicates that the first test code word is correct in response to being provided with the first test code word.

9. The method of claim 1, further comprising:

providing a set of test code words to the error correction/detection logic, wherein an error has been introduced into each of the test code words in the set by substituting check bits corresponding to an unused syndrome for a correct set of check bits within each test code word, wherein each test code word comprises substituted check bits corresponding to a different unused syndrome than each other test code word in the set of test code words;

in response to said providing, the error correction/detection logic decoding each test code word in the set of test code words; and

verifying that the error correction/detection logic correctly identified the error in each of the test code words.

10. The method of claim 9, wherein said providing a set of test code words comprises creating an initial code word, wherein each bit in the initial code word has a same logical value.

11. The method of claim 10, wherein the same logical value is a logical 0, and wherein the substituted check bits in each code word equal a set of bits in one of the unused syndromes.

12. The method of claim 10, wherein the same logical value is a logical 1, and wherein each bit in the substituted check bits in each code word equals an inverse logical value of each bit in one of the unused syndromes.

13. A computer readable medium comprising program instructions computer-executable to:

create an initial data bit combination having n bits, wherein each data bit in the initial data bit combination has a same logical value as each other data bit in the initial data bit combination;

shift a first bit having an different logical value than the same logical value across the initial data bit combination, wherein each time the first bit is shifted, one of n data bit combinations is generated;

provide each of the n data bit combinations to error detection/correction logic;

compare a set of check bits generated by the error correction/detection logic with a known correct set of check bits for each of the n data bit combinations; and

dependent on an outcome of said comparing, generate an indication of whether the error detection/correction logic correctly generated the set of check bits.

14. The computer readable medium of claim 13, further comprising program instructions computer-executable to:

provide a set of m+1 test code words to the error correction/detection logic, wherein each code word has m bits, wherein a first test code word in the set of m+1 test code words is a correct code word, wherein each test code word other than the first test code word comprises a single-bit error at a different bit position within the code word than each other test code word; and

verify that the error correction/detection logic correctly decoded each of the m+1 test code words.

15. The computer readable medium of claim 14, wherein each bit in the first test code word has a same logical value, and wherein each test code word other than the first test code word comprises a bit having an opposite logical value at the different bit position.

16. The computer readable medium of claim 15, further comprising program instructions computer-executable to shift the bit having the opposite logical value across the first test code word, wherein each time the bit is shifted, one of the test code words other than the first test code word is created.

17. The computer readable medium of claim 14, further comprising program instructions computer-executable to verify that the error correction/detection logic detects each of the single-bit errors.

18. The computer readable medium of claim 15, further comprising program instructions computer-executable to verify that the error correction/detection logic corrects each of the single-bit errors that occurs in a data bit within each of the m+1 test code words.

19. The computer readable medium of claim 14, further comprising program instructions computer-executable to verify that the error correction/detection logic indicates that the first test code word is correct in response to being provided with the first test code word.

20. The computer readable medium of claim 13, further comprising program instructions computer-executable to:

provide a set of test code words to the error correction/detection logic, wherein an error has been introduced into each of the test code words in the set by

substituting check bits corresponding to an unused syndrome for a correct set of check bits within each test code word, wherein each test code word comprises substituted check bits corresponding to a different unused syndrome than each other test code word in the set of test code words; and

verify that the error correction/detection logic correctly identified the error in each of the test code words.

21. The computer readable medium of claim 20, further comprising program instructions computer-executable to create an initial code word, wherein each bit in the initial code word has a same logical value.

22. The computer readable medium of claim 21, wherein the same logical value is a logical 0, and wherein the substituted check bits in each code word equal a set of bits in one of the unused syndromes.

23. The computer readable medium of claim 21, wherein the same logical value is a logical 1, and wherein each bit in the substituted check bits in each code word equals an inverse logical value of each bit in one of the unused syndromes.

24. A tester for testing error correction/detection logic, the tester comprising:

test check bit generating means for creating a set of test data bit combinations and providing the set of test data bit combinations to a check bit generator comprised in the error correction/detection logic, wherein the set of test data bit combinations comprises n n-bit data bit combinations, wherein each possible logical value of each data bit is present in at least one of the n n-bit data bit combinations in the set of test data bit combinations;

comparison means for comparing check bits output by the error correction/detection logic for each of the n n-bit data bit combinations in

the set of test data bit combinations to known correct check bits for each of the n n-bit data bit combinations; and

indication means for generating an indication as to whether the check bits output by the error correction/detection logic are correct based on an output of the comparison means.

25. The tester of claim 24, further comprising:

test code word generating means for creating a set of test code words and providing the set of test code words to the error correction/detection logic, wherein the set of test code words comprises m+1 m-bit test code words, wherein a first test code word is a correct code word, and wherein each other test code word comprises a single-bit error, and wherein each of the other test code words comprises the single-bit error at a different bit position than any other of the other test code words;

wherein the comparison means are further for comparing an output of the error correction/detection logic with a known correct output for each test code word in the set of test code words.

26. The tester of claim 24, further comprising:

test code word generating means for creating a set of test code words and providing the set of test code words to the error correction/detection logic, wherein an error has been introduced into each of the test code words in the set by substituting check bits corresponding to an unused syndrome for a correct set of check bits within each test code word, wherein each test code word comprises substituted check bits corresponding to a different unused syndrome than each other test code word in the set of test code words;

wherein the comparison means are further for comparing an output of the error correction/detection logic to a known correct output for each of the test code words.

27. A method of testing error correction/detection logic, the method comprising:

providing a set of m+1 test code words to the error correction/detection logic, wherein each code word has m bits, wherein a first test code word in the set of m+1 test code words is a correct code word, wherein each test code word other than the first test code word comprises a single-bit error at a different bit position within the code word than each other test code word;

in response to said providing, the error correction/detection logic decoding the set of m+1 test code words; and

verifying that the error correction/detection logic correctly decoded each of the m+1 test code words.

28. The method of claim 27, wherein each bit in the first test code word has a same logical value, and wherein each test code word other than the first test code word comprises a bit having an opposite logical value at the different bit position.

29. The method of claim 28, wherein said providing comprises shifting the bit having the opposite logical value across the first test code word, wherein each time the bit is shifted, one of the test code words other than the first test code word is created.

30. A method of testing error correction/detection logic, the method comprising:

providing a set of test code words to the error correction/detection logic, wherein said providing comprises introducing an error into each of the test code

words in the set by substituting check bits corresponding to an unused syndrome for a correct set of check bits within each test code word, wherein each test code word comprises substituted check bits corresponding to a different unused syndrome than each other test code word in the set of test code words;

in response to said providing, the error correction/detection logic decoding each test code word in the set of test code words; and

verifying that the error correction/detection logic correctly identified the error in each of the test code words.

31. The method of claim 30, wherein said providing comprises creating an initial code word, wherein each bit in the initial code word has a same logical value.

32. The method of claim 31, wherein the same logical value is a logical 0, and wherein the substituted check bits in each code word equal a set of bits in one of the unused syndromes.

33. The method of claim 31, wherein the same logical value is a logical 1, and wherein each bit in the substituted check bits in each code word equals an inverse logical value of each bit in one of the unused syndromes.

34. A data processing system comprising:

a storage array comprising at least one storage device;

a host computer system coupled to provide data to the storage array; and

error correction/detection logic configured to generate check bits for the data being provided to the storage array;

wherein the host computer system is configured to test the error correction/detection logic by providing each of a set of n data bit combinations to the error detection/correction logic, wherein each data bit combination has n bits, wherein each possible value of each data bit is present in at least one of the n data bit combinations, wherein the set of n data bit combinations provided to the error detection/correction logic is a subset of a set of all data bit combinations that it is possible to create using n bits;

wherein in response to being provided with the set of n data bit combinations, the error detection/correction logic is configured to generate a set of check bits for each of the n data bit combinations; and

wherein the host computer system is configured to compare the set of check bits generated by the error correction/detection logic with a known correct set of check bits for each of the n data bit combinations.

35. A method of testing error detection/correction logic, the method comprising:

providing a subset of possible data bit combinations of n data bits to the error detection/correction logic, wherein the subset comprises n data bit combinations, wherein each possible value of each data bit is present in at least one of the n data bit combinations in the subset;

verifying the error detection/correction logic by comparing a set of check bits generated by the error detection/correction logic for each of the n data bit combinations in the subset with a set of known correct check bits;

providing a first set of m+1 test code words to the error detection/correction logic, wherein a first test code word is a correct test code word and where each

other test code word in the set of m+1 test code words comprises a single-bit error, wherein each test code word having a single-bit error has the single-bit error at a different bit position than each other test code word that has a single-bit error;

wherein said verifying further comprises comparing a first output of the error detection/correction logic generated in response to said providing a first set of m+1 test code words with a first known correct output for each of the m+1 test code words;

providing a second set of test code words to the error detection/correction logic, wherein each test code word in the second set comprises an error introduced by substituting check bits corresponding to an unused syndrome for a correct set of check bits within a correct code word, wherein each test code word in the second set comprises substituted check bits corresponding to a different unused syndrome than each other test code word in the second set of test code words;

wherein said verifying further comprises comparing a second output of the error detection/correction logic generated in response to said providing a second set of test code words with a second known correct output for each of the m+1 test code words; and

in response to said verifying, indicating whether the error detection/correction logic is operating properly.